(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 March 2002 (28.03.2002)

PCT

(10) International Publication Number
WO 02/25482 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/CH01/00574

(22) International Filing Date: 21 September 2001 (21.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/234,124 21 September 2000 (21.09.2000) US

(71) Applicant (for all designated States except US): UNIVERSITAET ZUERICH [CH/CH]; Raemistrasse 71, CH-8006 Zuerich (CH).

(72) Inventors; and
(75) Inventors/Applicants (for US only): MAZZOLA, Guerino [CH/CH]; Bettlistrasse 42, CH-8600 Duebendorf (CH). GOELLER, Stefan [DE/CH]; Frohburgstrasse 192, CH-8057 Zuerich (CH).

(74) Agent: UNITECTRA AG; Schoenberggasse 2, CH-8001 Zuerich (CH).

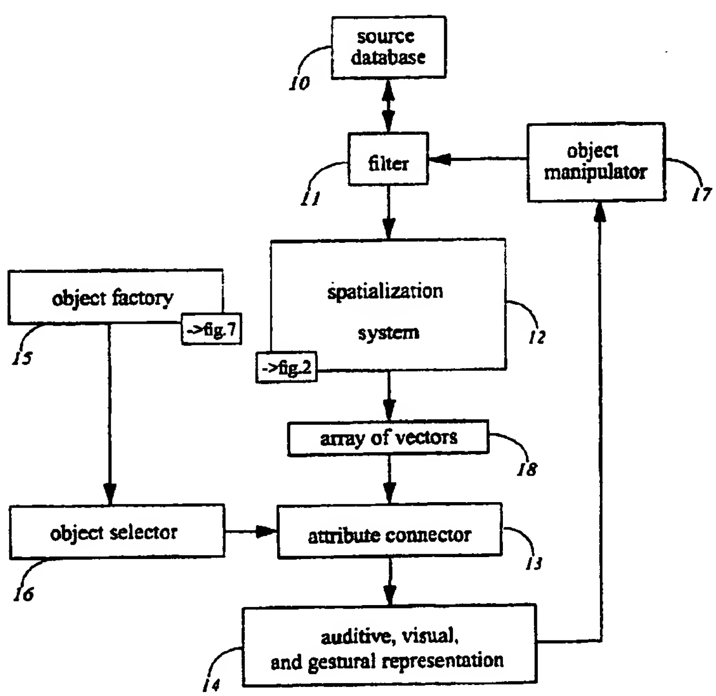
(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: PROCESS AND COMPUTER SYSTEM FOR VISUALIZATION OF DATABASE CONTENTS



(57) Abstract: In a process for visualization of database contents in a computer system, one selects a source database containing a predetermined type of data stored in a source format on a source information storage medium connected to the computer system and transfers the data contained in the source database to the computer system. A denotator-type filtering is applied to said transferred data to produce a database of linearly ordered denotators containing data denotators corresponding to said source format of said data. An order-conversing transformation procedure is then applied to the denotator database, thus producing an array of linearly ordered n-dimensional data vectors, wherein the linear order of data vectors corresponds to the linear order among the data denotators. The array of ordered data vectors is mapped onto a plurality of display objects, each vector being assigned a corresponding display object, each vector component being assigned an attribute of the corresponding display object, and each value of a given vector component being assigned a value of the corresponding attribute of the corresponding display object. Finally, the plurality of display objects is displayed.

WO 02/25482 A2

Process and computer system for visualization of database contents Technical Field of the Invention The present invention relates to a process for visualization of database contents, a computer system for visualization of database contents, a computer program product, a display object and a use of the computer system.

Prior Art The concept of knowledge space has become a popular metaphor for the contents of data- bases. Various types of database structures and management systems therefor have been described, for example, in US 5,201,046 (Goldberg et al.) and US 5,295,261 (Simonetti). In general, however, knowledge space is not realized as the familiar, three-dimensional geomet- ric space wherein database entities can easily be visualized as geometric objects. Rather than that, one is faced with the task of visualizing entities of an abstract higher dimensional space.

A central problem with navigation on and orientation in knowledge space is the necessity to implement order relations. In the special case of libraries and encyclopedias, the ordering is based on the alphabetic or lexicographic ordering. Several approaches to the construction of spatial orientation paradigms are known (for example: Hearst, M. and Karadi, C.:"Cat-a- Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category [HIERARCHY"; PROCEEDINGS OF THE 20TH ANNUAL INTERNATIONAL ACM/SIGIR CONFE-] rence, Philadelphia, PA, July 1997), but these are restricted to special databases such as li- braries or hypertexts. However, a generic visualization of database contents should allow for visual order representations that are independent of specific database configurations.

Multimedia systems offer the advantage of additional display means, that is, further percep- [TIVE] dimensions (such as sound) are added to the conventional pictorial display as available on a monitor screen. In the following, the term'visualization'will be taken to involve any type of display means including pictorial and non-pictorial (e. g. acoustical) display means. It ap- pears that in those instances where multimedia systems have been used for implementing visualization of databases, the spatial character of database entities was given a priori or was deducible in a straightforward manner from numerical parameter spaces. For example, IBM's Visualization Data Explorer is built on the data model for scientific visualization by Haber, R., Lucas, B. and Collins, N.:"A Data Model for Scientific Visualization with Provisions for Regu- lar and Irregular Grids" ; in: Proceedings of the IEEE Visualization'91 Conference, pp. 298- [305,] October 1991. However, this model is based upon visual units of geometrical nature. If a database does not consist of geometrical entities, visualization could only be realized after applying a specific ad hoc geometrical interpretation of abstract data.

Visualization of space configurations in more than three dimensions has not been dealt with except for a rather small number of additional dimensions. For example, the software'Mine- Set'by SGI implements the Splat Visualizer for objects in up to nine dimensions. In addition to the three familiar space dimensions, further dimensions can be attributed to specific parame- ters of visual objects, for example by resorting to color, shape and further properties of the visual objects employed. However, this method does not allow one to perform a non- destructive reduction of dimensions. By carrying out projections of higher dimensional spaces onto a three-dimensional space (e. g. familiar 3D-space) a plurality of coordinates is inevitably destroyed or lost.

As a further disadvantage, existing visualization systems lack appropriate tools for flexible and interactive construction and visualization of the multimedia-objects that are taken to rep- resent generic n-dimensional numeric data. For example, with the software'Maya'by [ALIAS/WAVEFRONT] one can interactively design 3D-objects and also position them in 3D-space, but neither interactive navigation nor direct manipulation of the object parameters is possible.

In summary, the prior art processes for visualization of abstract knowledge databases lack the following desirable features: (1) ordering paradigms that could be applied to an arbitrary type of database contents; (2) non-destructive reduction of spatial dimensions; (3) design and user interface principles for items (1) and (2) hereinabove that are universal, i. e. that are applicable to any database hierarchy; (4) tools for flexible and interactive construction of display objects suitable for visualization of higher dimensional numerical data.

Summary of the Invention It is an object of the present invention to provide an improved process for visualization of da- tabase contents.

It is a further object of this invention to provide an improved computer system for visualization of database contents.

It is yet another object of this invention to provide a computer program product for performing the process of visualization of database contents.

It is still another object of this invention to provide a display object produced by a process for visualization of database contents.

It is yet a further object of this invention to provide a use of the aforesaid computer system for interactively modifying the contents of a visualized source database.

The foregoing and further objects are achieved by the process defined in claim 1, the com- puter system defined in claim 7, the computer program product defined in claim 19, the dis- play object defined in claim 20 and the use defined in claim 21.

According to one aspect of this invention, a process for visualization of database contents in a computer system, comprises the steps of: a) selecting a source database containing a predetermined type of data stored in a source format on a source information storage medium connected to the computer system; b) transferring the data contained in the source database to the computer system; c) applying a denotator-type filtering to said transferred data to produce a database of line- arly ordered denotators containing data denotators corresponding to said source format of said data; d) applying to the denotator database an order-conserving transformation procedure which produces an array of linearly ordered n-dimensional data vectors, wherein the linear or- der of data vectors corresponds to the linear order among the data denotators; e) mapping the array of ordered data vectors onto a plurality of display objects, each vector being assigned a corresponding display object, each vector component being assigned an attribute of the corresponding display object, and each value of a given vector compo- nent being assigned a value of the corresponding attribute of the corresponding display object; and f) displaying the plurality of display objects.

According to a further aspect of this invention, a computer system for visualization of data- base contents comprises: a) selection means for selecting a source database containing a predetermined type of data stored in a source format on a source information storage medium connected to the computer system; b) transfer means for transferring the data contained in the source database to the com- puter system; c) filtering means for applying a denotator-type filtering to said transferred data to produce a database of linearly ordered denotators containing data denotators corresponding to said source format of said data; d) transformation means for applying to the denotator database an order-conserving trans- formation procedure which produces an array of linearly ordered n-dimensional data vec- tors, wherein the linear order of data vectors corresponds to the linear order among the data denotators; e) mapping means for mapping the array of ordered data vectors onto a plurality of display objects, each vector being assigned a corresponding display object, each vector compo- nent being assigned an attribute of the corresponding display object, and each value of a given vector component being assigned a value of the corresponding attribute of the cor- responding display object; and f) display means for displaying the plurality of display objects.

According to yet another aspect of the invention, a computer program product comprises pro- gram code means stored on a computer readable medium for performing the process of any one of claims 1 to 6 when the computer program product is run on a computer system.

According to still another aspect of the invention, a display object is produced by a [VISUALIZA-] tion process according to this invention.

According to yet a further aspect of this invention, the computer system of this invention is used for interactively modifying the contents of a visualized source database to produce a modified source database by applying the steps of: a) selecting one display object from a set of display objects produced from said visualized source database according to any one of claims 1 to 6; b) selecting an attribute of said selected display object; c) modifying the value of said attribute to produce a modified set of display objects; d) mapping the modified set of display objects to a modified array of ordered data vectors; e) applying to the modified array of ordered data vectors an order conserving transforma- tion procedure to produce a modified denotator database containing linearly ordered modified data denotators; and f) applying to the modified denotator database an operation

inverse of denotator-type fil- tering, resulting in producing the modified source database.

Advantageous embodiments of the invention are disclosed in the dependent claims.

There is an essentially unlimited number of display object types that could be used for visual- zation purposes. According to claim 2, the display objects are of a type selected from a plural- ity of display object types, which types previously have been generated by means of an inter- active generator device.

According to claim 3, the transformation procedure involves a plurality of decision steps that are controllable by means of an interactive input device. With this embodiment, the visualiza- tion process can be carried out interactively in a recursive way, if desired. According to claim 4, the plurality of decision steps includes a selection between either of a powerset denotator algorithm and single denotator algorithm. According to claim 5, the plurality of decision steps includes a selection of a value to be assigned to a display object greeking coordinate. Accord- ing to claim 6, the plurality of decision steps includes a selection of a vector component to be displayed either as a spatial coordinate or as another attribute of the corresponding display object. This is particularly useful when using multimedia environments.

According to claim 8, the selection means of the computer system include an interactive input device. According to claim 9, the computer system further comprises means for interactively generating a plurality of display object types.

According to claim 10, the transformation means include an interactive input device for input- ting a selection between either of a powerset denotator algorithm and single denotator algo- rithm. According to claim 11, the transformation means include an interactive input device for inputting a selection of a value to be assigned to a display object greeking coordinate.

According to claim 12, the transformation means include an interactive input device for inputting a selection of a vector component to be displayed either as a spatial coordinate or as another attribute of the corresponding display object.

In principle, the display means could be a simple display screen. Preferably, the display means include a multimedia display device as defined in claim 13, thus allowing displaying a plurality of display object attributes.

According to claim 14, the mapping means include multimedia software tools and an interac- tive input device [FCRASSIGNING] vector components to multimedia coordinates of the multimedia display device.

According to claim 15, the computer system further includes an interactive input device for defining a navigation position in a virtual space containing the plurality of display objects, whereby a condition of proximity between the navigation position and one of said plurality of display objects is interpreted as a selection of said one of said plurality of display objects.

According to claim 16, the computer system further includes means for modifying a visualiza- tion property of a selected display object.

According to claims 17 and 18, the computer system further includes means for displaying and storing, respectively, a navigation trajectory defined by a sequence of navigation posi- tions.

Brief Description of the Drawings The above mentioned and other features and objects of this invention and the manner of achieving them will become more apparent and this invention itself will be better understood by reference to the following description of various embodiments of this invention taken in conjunction with the accompanying drawings, wherein: FIG. 1 is a flowchart representation of a process for visualization of the contents of a source database in a computer system; FIG. 2 is a flowchart representation of the generic boolean ramification in the spatializa- tion step of the visualization process of FIG. 1, showing the splitting up into an algorithm for single denotators and an algorithm for sets of denotators; FIG. 3 is a flowchart representation of the algorithm for single denotators of FIG. 2, showing the ramification into five denotator types, each of which requires a separate spatialization routine; FIG. 4 is a flowchart representation of the algorithm for sets of

denotators of FIG. 2, showing the ramification into five referenced denotator types, each of which requires a separate spatialization routine; FIG. 5 is a flowchart representation of the algorithm for limit type denotators which is based on a-folding algorithm that reduces [DNNENSION,] wherein the reduction is recursive and is based on the 2D-folding algorithm; FIG. 6 is a flow chart representation of the 2D-folding algorithm which is based on a se- ries of compactification functions and yields a one-dimensional image of a set of 2D- vectors; FIG. 7 is a flow chart representation of a factory for construction of multimedia-objects; and wherein FIGS. 8 through 15 refer to the example given in the description.

The exemplifications set out herein are not to be construed as limiting the scope of this dis- closure or the scope of this invention in any manner.

Detailed Description of the Invention FIG. 1 illustrates a process for visualization of database contents in a computer system. One first selects the database to be visualized, henceforth referred to as'source database'. The source database 10 contains a predetermined type of data stored in a source format on a source information storage medium connected to the computer system. In practice, the stor- age medium could be a component of the computer system or it could be part of an external system connected to the computer system by a suitable data communication line, e. g. via a global communication network such as the Internet.

Subsequently, a filtering process 11 of the so called'denotator-type'as described in detail hereinbelow is applied to said transferred data to produce a database of linearly ordered de- notators; this database-henceforth called'denotator database'-contains data denotators corresponding to the source format of the data contained in the source database.

As a next step, one applies to the denotator database an order-conserving transformation procedure which produces an array 18 of linearly ordered n-dimensional data vectors, wherein the linear order of data vectors corresponds to the linear order among the data deno- tators. This transformation will henceforth be called'spatialization'.

Thereafter, the array of ordered data vectors is mapped onto a plurality of display objects ac- cording to the following procedure: to each vector is assigned a corresponding display object; to each vector component is assigned an attribute of the corresponding display object; and to each value of a given vector component is assigned a value of the corresponding attribute of the corresponding display object.

Finally, the plurality of display objects thus produced is displayed by appropriate display means. As will be discussed hereinbelow, the mapping and display process are preferably performed by means of an object factory 15, an object selector 16 and an attribute connector 13 so as to achieve, for example, an auditive, visual and gestural representation of the dis- play objects in a multimedia environment [14.]

It is understood that the data contained in the source database 10 can be transferred to the computer system by transfer means as known from the prior art. Moreover, the filtering [PROC-] ess, the order-conserving transformation (i. e. the spatialization) and the mapping process are carried out with suitable filtering means, transformation means and mapping means, respec- tively, each of which may comprise known computer hardware components controlled by computer software encoded for the filtering, transformation and mapping processes, respec- tively.

As shown schematically in FIG. 1 and described in more detail hereinbelow, the computer system may be used for interactively modifying the contents of a visualized source database to produce a modified source database by means of an object manipulator 17.

Denotator data model The denotator data model has previously been described in Zahorka, [O.] :"PrediBase-Controlling Semantics of Symbolic Structures in [MUSIC"; IN ICMC PROCEEDINGS ; (ICMA,] ed.), San Francisco 1995, and in: Mazzola, G. [:"MUSIC&COMMAT;ENCYCLOSPACE";] in: Proceedings of the [KLANGART] [CONFERENCE 1997 ;] (Enders, B., Stange-Elbe, J., Eds.), Rasch, [OSNABRUCK] 2000, the contents of which are incorporated herein by reference. Nevertheless, the model will be desc- ribed hereinbelow in a self-contained way.

In this data model, denotators stand for entities which are elements of domains, the latter being called forms. A

form NAME = TYPE<COORDINATOR> is defined by its NAME (a string), its TYPE (either Simple or Synonym or Limit or Colimit or Powerset), and its COORDINATOR, a sequence of forms or a mathematical module.

The Simple type defines the recursive root type and always takes a module M as its coordi- nator. In the context of this invention, it is supposed that M is provided with a linear order rela- tion, which is represented by means of the symbol < (not neccessarily compatible with the module structure). Examples for M are the Integer Numbers or the Real Numbers with their natural orderings; further examples for M are Integers modulo 26 with their induced ordering [0<1<2<...] <25, which are suitable for encoding the alphabet a, b, c,..., z.

The Synonym type has one single form F as its coordinator. Semantically, this encodes a re- naming of a given form.

The Limit type (which was called Product type in: Mazzola, G. loc. cit.) has a sequence [FI,] [..., FN] of forms as its coordinator. Semantically, this encodes a limit of a diagram of spaces in the sense of mathematical category theory. In the context of this invention, only the trivial dia- grams, i. e., cartesian products will be dealt with. Since general limits are subspaces of carte- sian products, their spatialization can be realized on subspaces.

The Colimit type (which was called Coproduct in: Mazzola, [G. LOC. CIT.)] has a sequence [FI,] ..., Fn of of forms as its coordinator. Semantically, this encodes a colimit of a diagram of spaces in the sense of mathematical category theory. In the context of this invention, only the trivial diagrams, i. e., coproducts (the analog of disjoint unions of sets) will be dealt with. Since general colimits are quotient spaces of [COPRODUCTS,] their spatialization can be realized by i- dentification of equivalent points.

The Powerset type has one single form F as its coordinator. Semantically, this encodes the powerset object of the given form in the sense of mathematical topos theory.

A denotator NAME = [FORM<COORDINATE>] is defined by its NAME, which is a string of characters, by its FORM, which is a form in the above sense, and by its COORDINATE, which is a datum that is defined according to the FORM's type as follows : FORM type = Simple : in this case, COORDINATE is an element of the FORM's module ; FORM type = Synonym : in this case, COORDINATE is a denotator of the FORM's COOR- [DINATOR] ; FORM type = Limit : in this case, COORDINATE is a sequence [DI,...,] [DN] of denotators in the COORDINATOR-forms [FOL,...,] Fn, respectively ; FORM type = Colimit : in this case, COORDINATE is a denotator in the COORDINATOR- form Fi at position i; FORM type = Powerset: in this case, COORDINATE is a finite set {D1, ..., Dn} of denotators in the COORDINATOR-form F.

For the set of denotators of a given form, a linear order has been defined in: Mazzola, G. [LOC.] [CIT..] This definition is recursive and extends the given linear order on the modules of simp- le forms. Example : Word = Colimit<Terminal, Unit> Unit = Limit<Letters, Word> Letters = [SIMPLE<Z26>] Terminal = [SIMPLE<Z1>] Z26 is the above module which encodes letters, [Z1] is the zero-module. The denotators of form Word (a circular form) are precisely the finite or infinite words built on the usual alphabet, and the resulting order relation on this form (defined in: Mazzola, [G. LOC. CIT.)] is the usual lexi- cographic ordering of words.

It should be pointed out that the source database 10 could be encoded on the source infor- mation storage medium in any of the known database formats, for example in the SQL or XML format. The denotator-type filtering 11 yields a denotator database D = F<C> in a speci- fic form F with a coordinator C depending on the source database, which can be represented, for example, in the [ASCII-BASED] context-free language denoteX (Noll, T. H., Garbers, J. :"Glo- bal Music Theory" ; in: Proceedings [OF THE KLANGART CONFERENCE 1999] ; (Rasch), Osnabruck, to appear in 2002) and parsed to denotator instances of certain Java classes. The Java pro- gramming language is encoded on a Java Virtual Machine that can be run on a commercially available computer system, for example on a SGI 320VW type computer system. Accordingly, the following algorithm can be implemented in the Java programming language, for example.

[SPATIALIZATION] Given D, the user interaction allows the input of a representation granularity in form of a non- negative integer, the modus [Y.] For example, this input can be implemented on a text field of a computer's user-interface window. The role of y will be to limit the recursive construction of spatialization beyond depth y of the denotator's recursive construction. Beyond this level, the spatialization algorithms will identify all intervening denotators and stop the spatialization pro- cess.

Given D, the user interaction allows the input of a folder matrix [.] The matrix coefficients are 0 or 1 and codify the user's preferences for grouping factor or cofactor forms in order to reduce their dimension (in case of limits) or arrange cofactor forms in 3-space (in case of limits). Its precise role will be explained hereinbelow. For example, a folder matrix can be implemented as a rectangular table of boolean buttons on a window of a user-interface. The matrix size can be set on a text field of the same user-interface window.

Given D, if its form F is of type Powerset, the user interaction allows the input of a boolean flag for powerset disclosure, for example implemented by a boolean button on a user- interface window. If the flag is set and the denotator's form F is of powerset type, the spatial- zation algorithm will produce an array which contains the spatialization of all denotators con- tained in D. This means that the user has the alternative to spatialize D as a single denotator or to disclose its elements.

The output of the spatialization operation 12 is an array 18 of vectors in n-space, represented as a Java instance, for example. Each of these vectors can be mapped to numerical [ATTRIBU-] tes of a multimedia-object which was selected in the object selector 16, where a multimedia- object has been taken from the object factory 15. For example, such a multimedia object can be represented als a Java instance and shown to the user as a graphical object in a window of the user-interface. In the attribute connector 13, the precise assignment of vector coordina- tes to object attributes is defined. for example, this can be implemented on the basis of drag-and-drop-methods for graphical user interaction. These data are represented on the auditive, visual, and [GESTURAL LEVEL] 14, for example on a SGI 320VW machine's screen and its audio equipment. The user may interact with these multimedia-objects via standard interaction de- vices, such as mouse, dataglove (for example by 5th Dimension [TECHNLOGIES),] joystick, or tracker technology (for example the tracker'Flock of Birds'by Ascension) and change the background denotator values via the object manipulator 17.

The spatialization system works as follows. FIG. 2 shows the initial boolean alternative 20 which is based on the powerset disclosure setting and can be implemented on a special but- ton of a user-interface window. In the case of Powerset = NO, spatialization is performed by the single denotator algorithm 22. However, this algorithm is only evoked if the greeking al- ternative 21 is set to NO. In the case of Greek = YES, the denotator is shown by 23 as a point of standard greeking coordinate (e. g. y = 0.5) on the real line. In the case Powerset = YES of disclosure 20, the powerset denotator algorithm 24 is evoked.

FIG. 3 shows the single denotator algorithm 22. This algorithm splits into five cases, 30,31, 32,33,34, according to the type of D's form F. Case 30 (Synonymy) evokes a modus decre- mentation 35 of the greeking modus [Y] and replaces the denotator D by its coordinate C. Case 31 (Simple) evokes a routine 36 which returns a real number f (F, C) as a function of the form F and the coordinate C. Such a routine in the simple algorihm must be implemented for each module and form name for simple forms. For example, if musical duration is represented by the simple form Quarters = [SIMPLE<INTEGERS>] then the denotator D = [QUARTERS<D>] can be mapped to the real number f (Quarters, d) = d/4 to indicate the effective duration.

Case 32 (Limit). Here, D = [(D1,...,] [DN),] and the single denotator algorithm can be applied re-cursively to each coordinate [DI] with a decremented greeking modus y-1 and with default [LXMI] folder matrices which data returns a real number (one dimensional vector) [DI] for each coordi- nate [DI.] For the limit denotator algorithm it is supposed that the given folder matrix of D has S rows and n columns, and has the following two properties : (a) Each column has exactly one 1 entry.

(b) If a column has its 1 entry at row k, the following column has its 1 entry at row larger or [EQUAL K.]

For each row t, the columns where the 1 entry is positioned in this row define a subspace of the n-dimensional space. The corresponding coordinates d [(J,] [I),...,] [D (I, R)] define a vector d (t). FIG.

5 shows the folding algorithm which is applied to transform d (t) to a real number [5] (t) and which will be described below. The limit denotator algorithm 37 returns the vector [8] = [(8] [(1), ...]

[8] (S)) in S-space.

Case 33 (Colimit). In this case, the coordinate of the denotator D is a denotator Dj in the coordinator form Fj, and the single denotator algorithm can be applied recursively to coordina- te Dj with a default [1XMJ FOLDER] matrix which returns a real number dj, using a decremented greeking modus y-1. In this case the folder matrix [#] of D has S rows and n columns, and must have exactly one 1 entry in each column. A maximal uninterrupted sequence s of columns such that the successor of such a column has its 1 entry on a row which is larger or equal to the preceeding 1 entry row is called a monotony sequence. The matrix [#] splits uniquely into a sequence [SI, . . . ,] [SK] of monotony sequences. If the position j of denotator Dj is in sequence [SE,] and if the 1 entry of column j is on row f, then the denotator is mapped to the three dimensio- [NAL] vector (dj, f, e).

Case 34 (Powerset). In this case, D=F<{D1, ..., Dn}> is a set of denotators D1, ..., Dn in the F-coordinator form G. By recursion, with a decremented greeking modus [Y-1,] the single denotator algorithm, with a default 1xm folder matrix returns n real numbers [DOL, . . . ,] dn for the [D-ELEMENTS D1, . . . , DN7] respectively. Let 8 be the average value of these real numbers. Then the 2D-folding algorithm FIG. 6 associates with the two dimensi- onal vector (1,8) a real number d as output.

FIG. 5 (folding algorithm). This routine transforms a finite array E = [(EI, . . . ,] et) of n- dimensional vectors ei=(x(i,1), ...,x(i,n)) to an array U = [(UT, . . . ,] ut) of real numbers in such a way that the lexicographic order relation between the n-dimensional vectors is mapped to the natural order relation among real numbers as follows : If n=1 (50), the output is identical to the input. If n=2 (51), the 2D-folding algorithm 52 (see next paragraph) yields a finite array of real numbers. If n>2 (53), the projection E'of E onto the last two coordinates is built, and the 2D- folding algorithm is applied. This yields an array of real numbers which are now inserted as new coordinates on dimension n-1 of the given vectors. Now recursion is applied to the new array of the [N-1-DIMENSIONAL] vectors.

FIG. 6 (2D-folding algorithm). This algorithm maps a finite array E = [(E1, . . . ,] et) of 2D-vectors e i [= (XJ, YJ)] to an array U = [(US, . . . ,] ut) of real numbers in such a way that the lexicographic or- der relation between the 2D-vectors is mapped to the natural order relation among real num- bers. Routine 60 builds a sequence of adjacent vertical strips in 2-space, each being defined by the middle between successive coordinates [XI, . . . ,] xt. The routine 61 uses a real parameter [A.] For each vertical strip, a bijective bicontinuous and order preserving function [CEP ;] (y) of com- pactification from the extended real line [[- OO,] [+OO]] onto the closed interval definded by the ho- rizontal limits of the strip is required (e. g. a function of type arctan (y)). The parameter a is mapped to the middle xi of the strip. Routine 62 applies each vector [EL] to [U ;] = cpi(yi).

In the YES-case of the powerset disclosure alternative 20 of FIG. 2, spatialization is perfor- med by the powerset denotator algorithm 24. Recall that this alternative is only possible if the denotator's form F is of type Powerset. This means that D = F< [{DI, ...] Dn} > is a set of deno- tators [DU, ...] Dn in the F-coordinator form G. The powerset denotator algorithm is described in the flowchart of FIG. 4. This algorithm splits into five cases, 40,41,42,43,44, according to the type of the common [DI] form G.

Case 40 (Synonymy) evokes a modus decrementation 45 of the greeking modus [Y] and repla- ces the denotators D1, ..., [DN] by their coordinates C1, ..., Cn, respectively and the total denota- tor D by the powerset denotator D'= G<{C1, ..., Cn}>.

If [Y = 0] from the beginning, the routine returns the standard greeking vector described in 23.

Case 41 (Simple) evokes the simple denotator 46 power algorithm which applies the single denotator simple algorithm to each element [DI,] yielding [UI,] and then returns the array of real numbers [UT, . . . ,] [UT.]

Case 42 (Limit) evokes the limit denotator power algorithm 47. Each element [DI] of [D] = F< {D1, ..., Dn} > is now an k-dimensional array of denotators [DJ =] (D i, [I) , . . . ,] D i, [K))] of forms [FOL, . . .] Fk, respectively. If [D&APOS;IS] the powerset denotator who's elements are the j-com- ponents of the denotators [DI] of D, [Y] is decremented by 1, and the powerset denotator [ALGO-] rithm is recursively applied to every denotator [DI,] yielding an array Uj = (u(j,1), ..., u(j,n)) with a default 1xmj folder matrix. This data defines an array U = [(UT, . . . ,] [UN)] of k-dimensional vectors Uj [=] (u(j,1), ..., u(j,k)). To each vector of this array the single denotator algorithm for the limit type is applied, starting from the situation of case 37 after the decrementation of the greeking mo- dus, and dealing with the given folder matrix.

Case 43 (Colimit) evokes the colimit denotator power algorithm 48. In this case, the denotator D's form is provided with a sequence F1, ..., Fk of coordinator forms. D is the disjoint union of its restrictions Dj = Fj<{D (j,1), ..., D(j,n(j))} > of denotators [DI] with position j. For each such restric- tion, [Y] is decremented by 1, and we obtain an array Uj=(u(j,1), ..., u(j,n(j))) of real numbers, by recursion and starting from a default 1xmj folder matrix. Retaking the process of the colimit routine 33, let [SI, . . . ,] sk be the sequence of monotony sequences. One considers the sequen- ce of the U'which pertain to a fixed monotony sequence se and row f. The interval algorithm described below yields an array a (e, f) = (a (e, fi),..., a (e, f, 1 (e, f))) of real numbers which correspond one-to-one, and order preserving, to the denotators of the restrictions D'which are indexed in the given monotony sequence se and have their 1 entry on row f.

To describe the interval algorithm, one considers the first row f in monotony sequence [S1] and its arrays [U1,] [U2, . . . ,] [UG.] The general case works on the same principle. The following array of 2D vectors is now available : Its vectors are of shape (i, U [(J,] [H)) ,] where 0 < h < n (i) [+1,] 0 < i < [G+1.] The 2D folding algorithm 83 is applied to this array and yields the array [A (I, F) .]

The denotator element [D ;] of D in monotony sequence se at row f which is associated with coordinate [A (E, F, R)] is given the 3D-vector [(A (E, F, R) ,] f, e).

Case 44 (Powerset) evokes the powerset denotator power algorithm 49. Here, the elements [DI] of D are powerset denotators of form G: Di = G< {D i, [1) , . . . ,] D i, [N] [(I)) }] > For each such element, the average [VALUE 5I] is calculated according to algorithm 34 for single powerset denotators. Since denotators of the common form F have a linear order, the ele- ments [D ;] of D can be ordered. One may suppose that the index i is the order index, i. e. [DI] < Dj iff [I<J.] This data produces an array of 2D-vectors (i, [S,) .] The 2D-folding algorithm 52 maps this array to an array of real numbers.

Multimedia [OBIECT] [FACTORV] FIG. 7 describes the flow chart of a process for generating three-dimensional display objects.

The output 74 of this process is a collection of 3D multimedia-object types implemented, for example, as a list of [JAVA3D] instances. The attributes of such display object types may typi- cally include spatial position, scale, color, shape parameters, rotation angles, animation pa- rameters, haptic specifications, sound parameters. These attributes are all freely accessible (for example by means of specific Java methods) in the computer system's runtime. Each 3D multimedia object type is generated via filter 73 (for example written as a Java routine) from a corresponding denotator which has been made available multimedia-object denotator editor 72. Such a denotator may be produced with this editor or else it may be loaded from denota- tor database 71. It should be noted that the denotators backing the 3D multimedia display ob- jects are described by the same data model (denotators and forms) as the source database to be visualized on this system.

The architecture of multimedia-object denotators follows a particular form which solves the problem of building compound multimedia objects with arbitrary recursive depth structure.

The form may typically be defined as follows : Sat = Limit<Geo, Satlist> [SATLIST] = Colimit<Terminal, [ETC>] ETC = Limit<Satlist, Sat> Geo = Limit<RTC, URL> URL = [SIMPLE<ASCII>] wherein ASCII is the free [Z-MODULE] over the monoid of words built from the ASCII symbols RTC = Limit<Trans, Rot, Scale, Color, Tex, Sound> [F = SIMPLE<ASCII>] Onset = [SIMPLE<R>] (R = real numbers) Duration = [SIMPLE<R>] t= Limit<Onset, Duration Trans = [LIMIT<F,] f, f, t> Rot = [LIMIT<F,] f, f, f, [T>] [SCALE = LIMIT<F,]

f, f, t> Color = [LIMIT<F,] f, f, [T>] Tex = [LIMIT<F,] [T>] Sound = [LIMIT<F,] f, f, f, f, t> The meaning of the above settings is the following. The Geo component describes the shape and the behavior of the multimedia display object referred to in the [URL-VARIABLE] and loaded from the 3D factory object database 70. The RTC component describes the multimedial be- havior of the URL object by a series of parameterized functions whose coefficients define the 3D display object's freely accessible variables. Each of the components Trans, Rot, etc., sha- res the time specification t in order to articulate onset and duration of this component.

The compound structure of this form's denotators is enabled by the circular Satlist compo- nent: Each Satlist specification refers to either a terminal value, in which case the recursion stops, or it refers to the ETC component which is again split into a Sat component for a further Geo denotator, and a Satlist for recursive prolongation. The latter refers to already de- fined denotators which can be referred to via the denotator database 71.

From the existing collection of 3D multimedia display objects, the user may now select one display object X in the object selector 16, for example by selecting a graphical representation of the object in the list shown on a user-interface window. In order to generate a collection of concrete multimedia display objects in 3D space, the attribute connector 13 is used to con- nect each attribute of X to the vector coordinates of the given vector array obtained form the spatialization routine 12. This procedure yields the multimedia representation 14 of the vector array, and therefore, of the source database 10.

Obiect manipulation As indicated in FIG. 1, the computer system may be used for interactively modifying the con- tents of a visualized source database in order to produce a modified source database by means of object manipulator 17. This application is based on the fact that the visualization process can be inverted. Assuming that source database 10 has been visualized by applying the visualization process described hereinabove, the contents of this visualized source data- base can be modified by first selecting one display object from a set of display objects pro- duced from said visualized source database. Subsequently, one selects an attribute of said selected display object and modifies the value of said attribute. This results in a modified set of display objects. Thereafter, the modified set of display objects is mapped to a modified ar- ray of ordered data vectors by a procedure inverse to the original mapping procedure of the visualization process. An order conserving transformation procedure inverse to the original spatialization procedure of the visualization process is then applied to the modified array of ordered data vectors, thus producing a modified denotator database containing linearly or- dered modified data denotators. Finally, an operation inverse of denotator-type filtering is ap- plied to the modified denotator database, resulting in producing the modified source data- base. For example, if the [ATTRIBUTE"COLOR"OF] a display object is the visual representation of the price of a certain item, which price is contained in the source database so visualized, then one can modify the color of the display object, e. g. by means of an interactive multimedia in- put device. The operational steps described hereinabove then result in a corresponding modi- fication of the corresponding price entry contained in the source database.

Further comments As will have become apparent, according to the user's preference for grouping and reducing of space dimensions, for granularity of representation, and for various types of multimedia display objects, the result of the visualization process may be chosen to be a collection of three-dimensional multimedia objects sharing attributes of shape, spatial position, auditive, animated, and interactive behavior. This result is in one-to-one correspondence with the enti- ties of the source database and preserves their linear ordering relations in the space of object attributes. The recursive design principles of the above described algorithms are applied in a uniform way to any hierarchical structure and depth of the particular source database.

The starting point for the visualization process is the representation of the external source da- tabase in the inner data model of denotators (the model's entities) and forms (the model's domains). The denotator model is recursively defined and uses all basic types of universal constructions in mathematical topos theory (limits, colimits, power objects). The root forms are mathematical modules (generalized vector spaces) which are provided with linear order relations. This data model also includes circular form definitions. By the recursive construction principle, such forms are canonically provided with linear orderings. This inner data model is general enough to allow all common database formats (e. g. entity relationship format, SQL, XML, denoteX) to be incorporated by adequate filter routines. This solves the task of provid- ing a universal and recursive database modeling, including canonically defined linear order relations.

Preferably, the user interface of the computer system for visualization of database contents includes various types of user defined specifications of the visualization process: a folder ma- trix, a granularity of

representation, a powerset disclosure, a display object factory, a display object selector, an attribute connector, and an object manipulator.

The folder matrix encodes the user's preferences for grouping factor or cofactor forms in or- der to reduce their dimension (in case of limits) or arrange cofactor forms in 3-space (in case of limits). This interface allows the user to reduce dimensions at will, thereby conserving the given linear order relations among the database's denotators when they are transformed into spatial output vectors.

In order to give the user control over the representation granularity of the spatial output, a greeking procedure analogous to text greeking in typesetting is applied beyond a selected search depth within the database ramification hierarchy. This granularity specification is par- ticularly important in order to deal with circular denotators which could otherwise produce infi- nite search loops.

Powerset disclosure causes the spatialization procedure to produce an array of n-dimensional output vectors which correspond to the contents of the powerset denotator. Powerset disclo- sure is a very frequent situation in the practical application of the invention.

The object factory is a toolbox for text-based and/or graphically interactive definition of com- pound multimedia display objects to be built from a plurality of predefined display object types. Such multimedia display objects share attributes which specify their spatial position, shape proportions, animated behavior, audio properties, haptic response and further proper- ties relevant for user interaction by means of known interactive devices such as mouse, data glove, joystick or tracker technology.

The object selector is a user-interface showing the arsenal of available display object types and allowing the user to select a particular type of display object for database visualization.

The attribute connector is a user-interface allowing the user to map the coordinates of the output vectors obtained from the spatialization procedure to numerical attributes of the se- lected display objects.

The object manipulator is a user-interface for editing the values of object attributes of any specific selection among the set of represented display objects by means of standard interac- tion devices.

The spatialization procedure is defined in such a way that whenever a number of denotators is transformed into a plurality of display objects 3D-space, then their original linear order rela- tionships are preserved on the output level in the natural lexicographic ordering of display ob- ject attributes. In particular, the display objects are a bijective image of the denotators of the source database.

The above means that the reduction of dimensions which is defined by the folder matrix, the spatialization granularity, and the spatialization procedure does not destroy the size and order relation of the source database. This meets the task of a non-destructive reduction of spatial dimensions.

The visualization process has been designed in a recursive way. Therefore, the visualization process can be applied in a uniform way to any hierarchical structure and depth of the source database.

The result of the spatialization procedure is an array of vectors in an n-dimensional space which can be mapped to a set of multimedia display objects in three-dimensional space.

Whenever desired, the display objects can be reconstructed and reselected by the user and connected in a different way to the abstract vector coordinates. They may also be acted upon by interaction devices in order to modify their attribute values.

Example The following illustration is a walk through a prototypical library whose contents are given as a database in denotator format. This example will be described on two levels, the first being an intuitive illustration, the second relating the intuitive description to the technical process dis- closed in the detailed description of the invention. Each step of the example will be described on the intuitive level first, and then on the technical level.

Suppose that the user has logged in on the homepage of a city X. He has identified the city's library and wants to navigate to particular documents. After a click on the library's link, the browser starts an implementation of the visualization system, for example a Java3D- Application ; in this example, we call this implementation the PrimaVistaBrowser.

As shown in FIG. 7, the first visualization of the library denotator on the PrimaVistaBrowser is an object in 3D-space. From the browser's object toolkit, an object is chosen having the shape of a library building as shown in FIG. 8.

Technical description : The library denotator is supposed to have powerset type. By default, the decision 20 in the detailed description is set to'powerset = no', and'greek = yes'. The generated object is placed at greeking coordinate 0.5 on an invisible line. The object's (library) graphical parameters are set to default values. If the user wishes, he can open a dia- log box to determine by a mouse click whether he wants to set the powerset to'yes'or 'no'. The user may also open a dialog box to set the greeking parameter [Y] to another value (this will be done in the next step!).

The user now approaches the building object shown in FIG. 8 by a mouse driven motion. As he gets in touch with the virtual building, the building'opens'and shows a vertical arrange- ment of three further objects as shown in FIG. 9, which are, from top to bottom: a text icon, an audio icon and an image icon. These objects represent cofactors in the denotator structure and their compact shape corresponds to greeking the library's contents below the top level of the colimit of three document types: print, audio, image. The space representation for these three document groups is generated by a preset folder matrix for the given colimit as de- scribed in case 43 of the detailed description.

Technical description: The approach action causes the decision 20 to switch to'powerset = [YES&APOS;.] This acti- vates the powerset denotator algorithm illustrated in FIG. 4. By default, the greeking mode in this subroutine is set [TO&APOS;=] 0'for each of the three cofactors. This means that the [COLIMIT] denotator power algorithm 48 of FIG. 4 [USES&APOS;=] 0'. Therefore, each deno- tator in any of these cofactors has the same greeking coordinate 0.5. The default folder matrix in this case is the codiagonal 3x3 matrix. The algorithm therefore yields three points on a line in 3-space as shown in FIG. 8. The graphical objects, which are positioned at these points (document, guitar, image) are chosen automatically but can also be changed by the user through appropriate selection from the browser's object toolkit.

Suppose the user wants to search for audio documents. He first zooms in to the audio icon of FIG. 9. To recognize the specific contents of the audio database, the granularity is refined. As shown in FIG. 10, the user now sees all the audio documents arranged on a vertical line in place of the original singular audio icon. In fact, the audio documents are denotators in a limit form. This means that their abstract representation as vectors is linearly ordered by a lexico- graphic total order in the database. This order is the same as the order on the line where these objects are positioned. The algorithms of this invention are defined in such a way that the visual representation always preserves the abstract order in the denotator database. In visual representation systems of the prior art, such a property is not guaranteed.

Technical description: The user now clicks on the desired object (in this case the audio icon). This causes the variable [Y] to be set to a nonzero value, either automatically or by using a dialog box. Subsequently, the colimit denotator power algorithm 48 (Fig. 4) is applied again, and the objects are now arranged according to the refined cofactor coordinates (befo- re, they were all set to 0.5).

The user now further approaches the vertical line arrangement of audio documents. These are by default represented by small audio symbols (CDs, Tapes). As the user is no longer in- terested in the overall 3D-coordinates of the space containing the vertical line, he decides to hide the space coordinates except for the coordinate along the vertical line. To this end, he calls [AN&APOS;OBJECTIVIZE&APOS;COMMAND] which collapses the space coordinates into a small gadget at- tached to the audio symbol and shown in FIG. 11 by a'smiley'symbol.

Technical description: From the colimit denotator power algorithm 48 of FIG. 4, the three space dimensions of the visible objects are such that the first two of them stem from the [CODIAGONAL] 3x3 matrix. Only the third coordinate gives deeper information about the single objects; in fact this coordinate was generated by the default folder matrix. For the objectivation of the two first coordinates the user chooses a graphical object from

the browser's object toolkit and assigns the two coordinates to the object's parameters. For the'smiley' face in our example, these are the degrees [OF&APOS;SMILE&APOS;AND&APOS;OVALNESS&APOS;.]

At this stage, the user could in principle approach each of the objects and thereby obtain a sample thereof. For example, if he approaches Beethoven's Fifth in Karajan's CD recording, the user would hear the famous motive at the beginning of this piece, as shown schematically in FIG. 12.

Technical description: The denotator linked to the present graphical object refers to audio data from the gi- ven database, which is loaded and played.

However, the properties of the audio documents could be made more explicit, since at this point they are still'hidden'in the linear arrangement of all these documents. By selecting (mouse click) the cofactor in question (audio documents), the visual representation speciali- zes to the cofactor of audio documents. On this level the folder matrix of the audio documents can be edited as indicated in FIG. 13. To this end the user opens the folder matrix, which de- fines the spatialization modus for the audio document subdenotator (cofactor). He decides that the spatialization should show the composer, [INSTRUMENTALIZATION] and duration, and that all other properties should be folded into one linear dimension. Whereas the first three coor- dinates are used to place the audio document in 3D, the last, folded, dimension is objectivi- zed as the audio object's color. To enter these settings, a window showing the folder matrix is opened, and the user may activate determined matrix positions by a mouse click. In FIG. 13, stars show the active matrix positions, whereas the rows can be assigned to space coordina- tes or object parameters (e. g. the object's color in the current example). So now, the user views a collection of colored audio symbols, each one still carrying its gadget from the above objectivization. As the user moves around in this virtual space, the three coordinates are shown on a small window such that the user can identify the object by its denotator data. For example, the composer coordinate shows the composer's name as the user approaches a specific audio icon.

Technical description: On the mathematical level, the audio document form is supposed to be of limit type, which means that an audio document is defined by a vector of denotator parameters.

The folder matrix shown in FIG. 13 is set in such a manner that the first three parame- ters'composer','instrumentation'and'duration'are associated with three numerical coordinates, which yield the 3D coordinates. In this process, the three denotator pa- rameters are mapped to the space coordinates by use of the default folder matrix for these three denotator parameters. All other denotator parameters are mapped to one numerical coordinate, related to row four of the folder matrix, by use of the folding al- gorithm of FIG. 5. The object which represents an audio document in this 3-space is the same as the one defined before selection of the audio document cofactor. To as- sociate the object's color attribute with the fourth row coordinate, the user opens an object inspector window, which shows the list of object attributes. By a drag-and-drop action from the folder matrix to the object inspector, the color attribute can be associ- ated with the fourth row coordinate.

Suppose the user has identified a document, has moved towards the corresponding object as shown in FIG. 14 and wants to know more about the folded coordinates and hide the three space coordinates representing'instrument','composer'and'duration'. To this end he objec- tivizes these three coordinates to object properties (shape, transparency,...). In turn, he spa- tializes the folded coordinates according to a new folder matrix whereby these coordinates enable new space coordinates (for example history, score and the other properties that were folded in one axis) and possibly additional object coordinates. The audio document at issue unveils further of its properties, which may be inspected through visual navigation. For exam- ple, the document's score may be approached and the collection of its notes can be popped up as a 3D configuration of new note objects.

In summary, the visualization tools are split into classical zooming ; objectivization; and [SPATIALIZATION] of any given coordinate selection.

A list of possible applications The visualization system can be integrated in a internet or intranet browser environment to represent sites and home pages in the described spatial way by means of filters from HTML/XML to Denotators. 'The visualization system can be used for spatially represented search tasks in connec- tion with search engines.

The visualization system can be loaded with graphically and/or acoustically and/or ges- turally interactive

editing tools of the denotators that are represented on the system's spatial level.

The visualization system allows for interaction with and not only for exploration of a given source database, i. e. the database may be modified.

A typical application of the modification feature deals with musical composition. Here, the original database would be a musical composition that is given from a music object data- base, e. g., a score file in a standard MIDI, KERN, MUSEDATA or denoteX format.

The visualization system can be used to enhance scientific visualization software in its frame navigation features, but also in detailed tasks for scientific visualization, sonifica- tion or [GESTURALIZATION.]

Description Claims

Claims [1.] A process for visualization of database contents in a computer system, comprising the steps of: a) selecting a source database containing a predetermined type of data stored in a source format on a source information storage medium connected to the computer system; b) transferring the data contained in the source database to the computer system; c) applying a denotator-type filtering to said transferred data to produce a database of linearly ordered denotators containing data denotators corresponding to said source format of said data; d) applying to the denotator database an order-conserving transformation procedure which produces an array of linearly ordered n-dimensional data vectors, wherein the linear order of data vectors corresponds to the linear order among the data de- notators; e) mapping the array of ordered data vectors onto a plurality of display objects, each vector being assigned a corresponding display object, each vector component be- ing assigned an attribute of the corresponding display object, and each value of a given vector component being assigned a value of the corresponding attribute of the corresponding display object; and f) displaying the plurality of display objects.

2. The process according to claim 1, wherein the display objects are of a type selected from a plurality of display object types, which types previously have been generated by means of an interactive generator device.

3. The process according to claim 1, wherein the transformation procedure involves a plu- rality of decision steps that are controllable by means of an interactive input device.

4. The process according to claim 3, wherein the plurality of decision steps includes a se- lection between either of a powerset denotator algorithm and single denotator algorithm. 5. The process according to claim 3, wherein the plurality of decision steps includes a se- lection of a value to be assigned to a display object greeking coordinate.

6. The process according to claim 3, wherein the plurality of decision steps includes a se- [LECTION] of a vector component to be displayed either as a spatial coordinate or as an- other attribute of the corresponding display object.

7. A computer system for visualization of database contents, the computer system com- prising: a) selection means for selecting a source database containing a predetermined type of data stored in a source format on a source information storage medium con- nected to the computer system; b) transfer means for transferring the data contained in the source database to the computer system; c) filtering means for applying a denotator-type filtering to said transferred data to pro- duce a database of linearly ordered denotators containing data denotators corre- sponding to said source format of said data; d) transformation means for applying to the denotator database an order-conserving transformation procedure which produces an array of linearly ordered n-dimensional data vectors, wherein the linear order of data vectors corresponds to the linear order among the data denotators; e) mapping means for mapping the array of ordered data vectors onto a plurality of display objects, each vector being assigned a corresponding display object, each vector component being assigned an attribute of the corresponding display object, and each value of a given vector component being assigned a value of the corre- sponding attribute of the corresponding display object; and f) display means for displaying the plurality of display objects.

8. The computer system according to claim 7, wherein the selection means include an in- teractive input device.
9. The computer system according to claim 7, further comprising means for interactively generating a plurality

of display object types.

10. The computer system according to claim 7, wherein the transformation means include an interactive input device for inputting a selection between either of a powerset denota- tor algorithm and single denotator algorithm.

11. The computer system according to claim 7, wherein the transformation means include an interactive input device for inputting a selection of a value to be assigned to a display object greeking coordinate.

12. The computer system according to claim 7, wherein the transformation means include an interactive input device for inputting a selection of a vector component to be dis- played either as a spatial coordinate or as another attribute of the corresponding display object.

13. The computer system according to claim 7, wherein the display means include a multi- media display device.

14. The computer system according to claim 13, wherein the mapping means include mul- timedia software tools and an interactive input device for assigning vector components to multimedia coordinates of the multimedia display device.

15. The computer system according to claim 13, further including an interactive input device for defining a navigation position in a virtual space containing the plurality of display ob- jects, whereby a condition of proximity between the navigation position and one of said plurality of display objects is interpreted as a selection of said one of said plurality of display objects.

16. The computer system according to claim 15, further including means for modifying a visualization property of a selected display object.

17. The computer system according to claim 15, further including means for displaying a navigation trajectory defined by a sequence of navigation positions.

18. The computer system according to claim 15, further including means for storing a navi- gation trajectory defined by a sequence of navigation positions. 19. A computer program product comprising program code means stored on a computer readable medium for performing the process of any one of claims 1 to 6 when the com- puter program product is run on a computer system.

20. A display object produced by a visualization process according to any one of claims 1 to 6.

21. Use of the computer system of any one of claims 7 to 18 for interactively modifying the contents of a visualized source database to produce a modified source database, char- acterized by the steps of: a) selecting one display object from a set of display objects produced from said visual- ized source database according to any one of claims 1 to 6; b) selecting an attribute of said selected display object; c) modifying the value of said attribute to produce a modified set of display objects; d) mapping the modified set of display objects to a modified array of ordered data vec- tors; e) applying to the modified array of ordered data vectors an order conserving trans- formation procedure to produce a modified denotator database containing linearly ordered modified data denotators; and [F)] applying to the modified denotator database an operation inverse of denotator-type filtering, resulting in producing the modified source database.

Description Claims